# 큐싱 공격 탐지를 위한 AutoML 머신러닝 기반 악성 URL 분류 기술 연구 및 서비스 구현

## (AutoML Machine Learning-Based for Detecting Qshing Attacks Malicious URL Classification Technology Research and Service Implementation)

김동영*, 황기성**
(Dong-Young Kim, Gi-Seong Hwang)

요 약

최근 정부 기관을 사칭한 가짜 QR(Quick Response)코드를 이용하여 개인정보와 금융정보를 탈취하는 QR코드와 스미싱을 결합한 '큐싱(Qshing)' 공격이 증가하는 추세이다. 특히, 이 공격 방식은 사용자가 단지 QR코드를 인식하는 것만으로 스미싱 페이지에 연결되거나 악성 소프트웨어를 다운로드하게 만들어 피해자가 자신이 공격당했는지조차 인지하기 어려운 특징이 있다. 본 논문에서는 머신러닝 알고리즘을 활용해 QR 코드 내 URL의 악성도를 파악하는 분류 기술을 개발하고, 기존의 QR 코드 리더기와 결합하는 방식에 관해 연구를 진행하였다. 이를 위해 QR코드 내 악성 URL 128,587개, 정상 URL 428,102개로부터 프로토콜, 파라미터 등 각종 특징 35개를 추출하여 데이터셋을 구축한 후, AutoML을 이용하여 최적의 알고리즘과 하이퍼파라미터를 도출한 결과, 약 87.37%의 정확도를 보였다. 이후 기존 QR코드 리더기와 학습한 분류 모델의 결합을 설계하여 큐싱 공격에 대응할 수 있는 서비스를 구현하였다. 결론적으로, QR코드 내 악성 URL 분류 모델에 최적화된 알고리즘을 도출하고, 기존 QR코드 리더기에 결합하는 방식이 큐싱 공격의 대응 방안 중 하나임을 확인하였다.

■ 중심어 : 큐싱 공격 ; 악성 URL ; 머신러닝 ; 사이버보안

**Abstract**

In recent trends, there has been an increase in 'Qshing' attacks, a hybrid form of phishing that exploits fake QR (Quick Response) codes impersonating government agencies to steal personal and financial information. Particularly, this attack method is characterized by its stealthiness, as victims can be redirected to phishing pages or led to download malicious software simply by scanning a QR code, making it difficult for them to realize they have been targeted. In this paper, we have developed a classification technique utilizing machine learning algorithms to identify the maliciousness of URLs embedded in QR codes, and we have explored ways to integrate this with existing QR code readers. To this end, we constructed a dataset from 128,587 malicious URLs and 428,102 benign URLs, extracting 35 different features such as protocol and parameters, and used AutoML to identify the optimal algorithm and hyperparameters, achieving an accuracy of approximately 87.37%. Following this, we designed the integration of the trained classification model with existing QR code readers to implement a service capable of countering Qshing attacks. In conclusion, our findings confirm that deriving an optimized algorithm for classifying malicious URLs in QR codes and integrating it with existing QR code readers presents a viable solution to combat Qshing attacks.

■ keywords : Qshing Attack ; Malicious URL ; Machine Learning ; Cyber Security

## Ⅰ. 서 론

정보화 시대에 들어와 이용자의 개인정보와 금융정보를 탈취하기 위한 고도화된 해킹 기법

* 학생회원, 경북소프트웨어고등학교 소프트웨어개발학과 김동영 (교신저자)
** 학생회원, 경북소프트웨어고등학교 인공지능소프트웨어학과 황기성

접수일자 : 2024년 02월 15일
수정일자 : 2024년 04월 24일
게재확정일 : 2024년 05월 07일
교신저자 : 김동영 e-mail : dongyoung0912@gmail.com

들이 증가하고 있다. 특히, 최근 QR코드를 활용한 스미싱 기법인 큐싱 공격이 수면 위로 떠 오르고 있다. 이용자들이 의심 없이 QR코드를 찍는다는 특징을 통해 가짜 QR코드를 정상 QR코드 위에 붙여 피싱 사이트로 유도하여, 금전적인 피해를 발생시키는 등의 큐싱 범죄가 이전보다 급격하게 증가하는 추세다 [1].

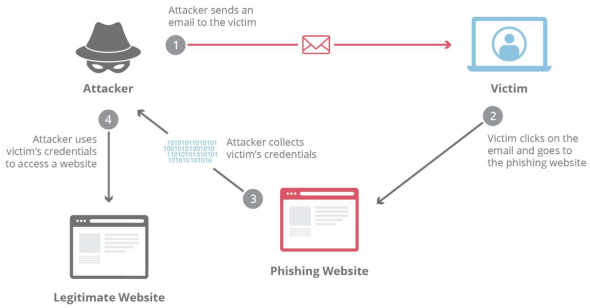


그림 1. 큐싱 및 스미싱 공격의 절차[2]

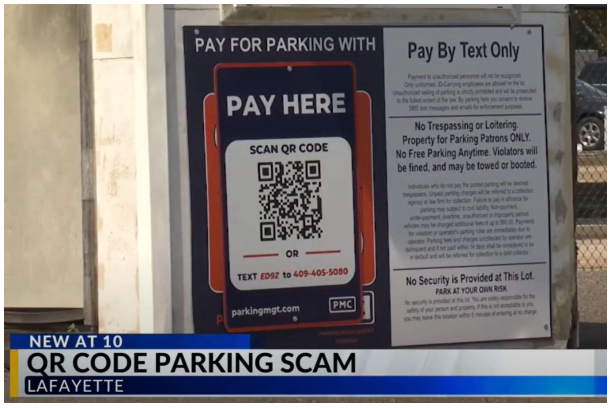


그림 2. QR코드를 활용한 실제 큐싱 공격[3]

이런 악성 QR코드와 악성 URL을 탐지하는 범용적인 탐지 방법으로는 지식 기반 탐지 기법을 사용하였다. 이전에 검증된 악성 URL 들을 블랙리스트에 저장하여 동일한 값의 저장 여부를 이용하여 탐지하는 방식을 지식 기반 탐지 기법이라 한다.

하지만 기존 지식 기반 탐지 기법의 지속적인 블랙리스트 DB를 업데이트 해야 하는 문제를 보완하고자 본 논문에서는 악성 URL에서 도메인, 호스트네임, 파라미터 등의 특징값들을 추출하고 머신러닝 기술을 활용하여 악성 URL을 분류하는 AutoML을 통해 최적화된 모델을 활용하여 학습시키는 것을 목표로 한다. 최종적으로는 학습시킨 악성 URL 탐지 모델을 기존의 QR코드 리더기에 결합한 서비스를 구현하고 테스트를 진행하는 순으로 해당 기술을 검증한다.

## Ⅱ. 선행 연구

2023년 6월 발표된 A Development of a Deep Learning-based Qshing DetectionApp using DGA and WHOIS Information[4]에서는 악성 도메인을 파악하기 위해 두 가지 방식을 사용한다. 첫 번째로, 이미 알려진 도메인은 지식 기반 탐지를 위해 검색 엔진이 빠른 mongoDB에 저장되어 판별한다. 두 번째로, DGA(Domain Generation Algorithm)로 생성된 도메인은 DGA 기계학습 모델을 사용하여 판별하고, 그렇지 않은 도메인은 WHOIS 기계학습 모델을 사용하여 악성 여부를 판별한다.

선행 연구에서는 DGA와 WHOIS를 통해 머신러닝으로 악성 URL을 분류하는 기술을 제안하였다. 하지만 이러한 탐지 방식의 경우 최근 새롭게 증가하는 무작위적인 악성 URL의 형태와 유지 시간, 지식 기반 탐지의 제한적이라는 한계가 존재한다.

본 논문에는 악성 URL과 정상 URL의 형태적인 차이점을 부분별로 대조하여 차이점을 파악하고, 해당 데이터를 기반으로 AutoML 기술로 악성 URL 분류 모델을 추출하여 실서비스에 적용한다.

## Ⅲ. Machine Learning Model

해당 문제는 QR코드에 내장되어있는 URL을 큐싱 공격인지 일반적인 URL인지 분류하는 이진 분류 문제이다. 해당 문제를 해결하기 위해

일반적인 URL, 큐싱 공격 URL로 구성되어있는 데이터를 사용, 지도 학습 머신 러닝 모델을 사용해 분류한다.

수많은 분류 머신 러닝 모델이 있지만, 해당 데이터에 가장 적합한 모델을 찾기 위해서 AutoML 라이브러리를 사용해 총 14개의 머신 러닝 모델과 튜닝된 모델 중 자동으로 해당 데이터에 최적화된 머신 러닝 모델을 도출한다. 사용된 모델의 종류는 8개가 사용됐으며 K-최근접 이웃, LightGBM, XGBoost, NeuralNet, CatBoost, WeightedEnsemble, ExtraTrees, RandomForest가 사용되었다. 이 중 눈여겨볼 모델은 RandomForest, ExtraTrees, Weighted Ensemble이다.

1. RandomForest

RandomForest는 분류 문제에 많이 사용되는 머신 러닝 알고리즘으로 하나의 출력을 하기 위해 여러 Decision Tree의 출력을 결합하여 출력을 결정한다. 이 알고리즘은 분류, 회귀 등 다양한 문제를 해결할 수 있지만, 악성 URL 탐지는 분류 문제이기 때문에 Randomforest를 분류기로 사용한다.

$$\phi_{B(x)} = Mode\phi(x, L_b) \quad (1)$$

$x$ : 모델에 입력되는 개별 데이터 포인트

$L_b$ : 개별 모델 또는 결정 트리

2. ExtraTrees

ExtraTrees는 Randomforest와 유사하게 앙상블 기법을 사용하는 머신러닝 알고리즘으로 Randomforest보다 더 큰 무작위성을 갖고 무작위로 각 노드에 임곗값을 선택함으로써 훈련 속도를 개선했다.

3. WeightedEnsemble

WeightedEnsemble은 여러 다른 모델의 예측에 가중치를 부여하여 결합하는 앙상블 기법으로 앙상블을 구성하는 각 모델의 예측 성능에 기반한 가중치가 부여된다. 높은 성능을 보여준 모델의 영향은 커지고 낮은 성능을 보여주는 모델의 영향은 적어진다.

$$\hat{p}(c) = \sum_{i=1}^{N} w_i \cdot p_i(c) \quad (2)$$

$w_i$ : $i$번째 모델의 가중치

$p_i(c)$ : i번째 모델이 데이터 포인트가 클래스 $c$에 속할 것으로 예측한 확률

## Ⅳ. 악성 URL 탐지 모델

1. 데이터 수집

본 논문에서는 정상 또는 악성 URL 데이터를 수집하고자 Kaggle에서 제공하는, Malicious URLs Dataset[5] 에서 정상 URL 428,102개와 한국인터넷진흥원에서 제공하는 한국인터넷진흥원_피싱 사이트 URL[6]에서 악성 URL 128,587개를 추출하여 하나의 데이터셋으로 구축하였다. 또한, 악성 URL에는 Label 1을, 정상 URL에는 Label 0을 Classification 학습을 위해 할당하였다.

2. URL 특징 추출 및 분석

악성 URL과 정상 URL 사이의 상관관계를 파악하고자 URL의 구성요소를 기반으로 하여 각 특징을 추출하고자 하였다. [7] URL의 스킴을 나타내는 프로토콜(Protocol), 호스트네임(Hostname), 위치를 지정하는 경로(Path), 파라미터(Params), 쿼리 문자열을 분석하여 키-값 쌍으로 분류하여 나타낸 쿼리 파라미터(Query_Params), 프래그먼트(Fragment)를 추출하여 URL을 구성하고 있는 구성요소를 기반으로 기본적인 데이터셋을 구축하였다.

표 1. URL 추출 특징 종류

| 분류 | URL 특징 | | |
|---|---|---|---|
| URL 형태 | Protocol 종류 | Hostname | Path 경로 |
| | Params 값 | Query_Params | Fragment 값 |
| URL 복잡도 | 특수 문자 종류 및 개수 값 | URL 길이 | URL 바이트 |

위 분석의 결과를 기반으로 ML 학습을 위한 데이터셋은 URL의 형태인 Protocol, Hostname, Path, Params, Query_Params, Fragment를 사용하며, URL의 복잡도를 나타내는 URL 내에 포함된 특수 문자('!@#$%^&*()-_+=[]{}|;:,.<>?/')의 개수, URL 문자열의 길이와 UTF-8로 인코딩하였을 때의 바이트 크기를 추가 특성으로 사용하였다.

## 3. 학습 데이터셋 분석

URL의 형태와 복잡도를 기준으로 하여 추출한 Feature 값들을 기반으로 하여 Matplotlib를 사용해 데이터 분석을 진행하였다.

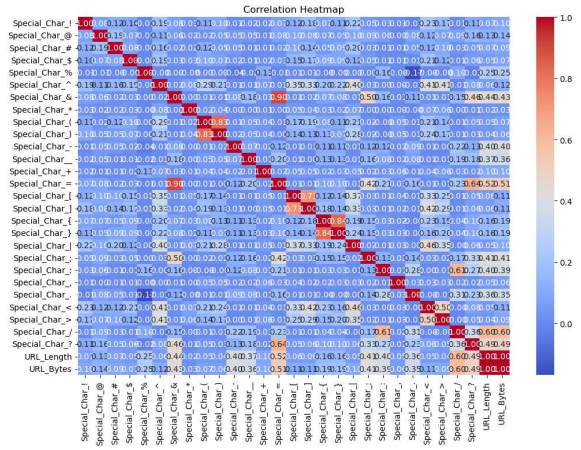


그림 3. 학습 데이터셋 Correlation Heatmap

그림 3과 같은 결과를 도출했으며 1이나 -1 즉 빨간색에 가까울수록 해당 Feature가 선형적이며 Feature가 0, 파란색에 가까울수록 비선형 Feature라는 것을 알 수 있다. 예를 들어 Feature Special_Char_& 와 Special_Char_= 는 0.9라는 높은 상관관계를 가지고 있다고 해석할 수 있다.

위 과정을 통해 URL의 복잡도를 기반으로 분석을 진행한 결과, 악성 URL과 정상 URL의 형태적 차이점을 파악하였다. 악성 URL의 경우 정상 URL과 달리 Protocol을 SSL 인증서를 사용하는 HTTPS보다 HTTP를 다수 사용하였으며 Params와 Query Params의 값에 타 URL 또는 불특정 메시지가 함께 포함되어 있는 경우가 분석한 URL 중 76% 이상으로 제일 많았다. 특히 URL의 복잡도를 기준으로 대조한 결과, URL 내에 포함된 특수 문자('!@#$%^&*()-_+=[]{}|;:,.<>?/')의 개수가 정상 URL 대비 악성 URL이 4배 이상 많았다.

표 2. 학습 데이터셋 구조 예시

| 분류 | URL 특징 | | |
|---|---|---|---|
| URL 형태 | Protocol (http / https) | Hostname (abc.com) | Path (/path) |
| | Params (Value) | Query_Params {key : Value} | Fragment (section) |
| URL 복잡도 | Special_Char_! | URL_Length (123) | URL_Bytes (125) |

우리는 URL의 형태적 특징과 특수 문자와 같은 복잡도를 기준으로 하여 악성 URL을 효과적으로 탐지하고자 AutoML 학습을 위한 학습 데이터셋을 표 2와 같이 생성하였다.

## 4. AutoML 머신러닝 학습

본 논문에서는 다양한 머신러닝 알고리즘을 이용해 데이터셋을 학습시켜 결과를 비교하고자 AutoML을 사용하였다. AutoML은 머신 러닝 모델을 쉽고 빠르게 구축할 수 있도록 도와주며 고성능 머신 러닝 모델을 학습시킬 수 있는 API를 제공하여 높은 성능의 모델을 빠르게 생성한다. 특히, 데이터 전처리 과정에서부터 알고리즘 선택 및 하이퍼파라미터 튜닝까지 모든 과정에서 품질 좋은 모델을 추출할 수 있으며[8], 최적의 모델을 찾고자 반복하는 학습 과정을 최소화

할 수 있다는 장점을 보유하고 있다. 하지만 모델의 상세한 하이퍼파라미터를 확인할 수 없다는 단점이 존재한다. [9]

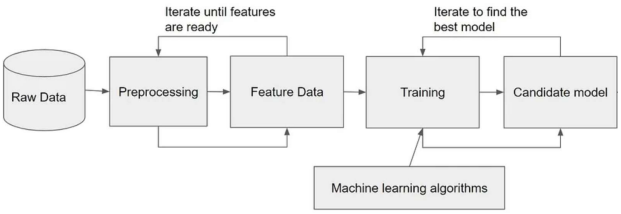


그림 4. AutoML Process

## Ⅴ. AutoML 실험 결과

악성 URL의 특징들을 추출하여 만든 학습 데이터셋과 AutoML을 사용해 학습을 진행하고 정확도, F1-Score를 사용해 성능을 평가하였다.

평가 지표를 F1-Score로 사용했을 경우, 테스트 기준으로 RandomforestEntr 가 87%, 정확도를 평가 지표로 사용했을 경우, RandomForest Entr 가 91%로 해당 실험에서는 제일 우수한 성적을 보여주었다. 다른 논문[10]에서 보여주었던 Accuracy는 97.00, 98.57 와 같았었다. 해당 논문에서는 전처리 방법을 통한 성능 개선을 시도하지 않았다. 그 차이로 모델의 성능에 차이가 생겼다.

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \qquad (3)$$

$H(X)$: 확률 변수 X의 엔트로피
$n$: 가능한 모든 사건의 수
$p(x_i)$: 사건 $x_i$가 발생할 확률

RandomForestEntr는 RandomForest에 엔트로피를 추가해 불확실성을 수치화한 값을 추가한다. 엔트로피를 추가함으로써 RandomForest 모델의 예측에 대한 불확실성을 이해하고 더욱 정보에 기반한 결정을 내릴 수 있게 한다.

엔트로피는 식3과 같이 계산되며 ExtraTrees 또한 엔트로피를 사용한 것이 사용하지 않거나 다른 함수를 추가하는 것보다 도움이 된다는 모델 학습 실험 결과를 도출하였다.

표 3. AutoML 학습 모델 결과 (Valid)

| Num | Model Name | Accuracy Valid | F1_Score valid |
|---|---|---|---|
| 1 | RandomForestEntr | 91.2663 | 85.7269 |
| 2 | RandomForestGini | 91.2663 | 85.6602 |
| 3 | ExtraTreesEntr | 91.2663 | 85.6881 |
| 4 | ExtraTreesGini | 91.2663 | 85.7440 |
| 5 | LightGBMLarge | 91.2887 | 85.3322 |
| 6 | WeightedEnsemble_l2 | 91.2887 | 86.0284 |
| 7 | XGBoost | 91.1540 | 85.4377 |

표 4. AutoML 학습 모델 결과 (Test)

| Num | Model Name | Accuracy Test | F1_Score Test |
|---|---|---|---|
| 1 | RandomForestEntr | 91.4322 | 87.3792 |
| 2 | RandomForestGini | 91.4322 | 87.3378 |
| 3 | ExtraTreesEntr | 91.4322 | 87.3777 |
| 4 | ExtraTreesGini | 91.4322 | 87.3778 |
| 5 | LightGBMLarge | 91.3172 | 86.7246 |
| 6 | WeightedEnsemble_l2 | 91.3172 | 87.2336 |
| 7 | XGBoost | 91.1737 | 86.6182 |

## Ⅵ. QR Reader 결합 시스템

악성 URL 탐지 모델을 기존 QR Reader와 연계하는 방식을 통하여 본 논문에서 개발된 악성 URL 탐지 모델의 성능과 실제 서비스에 도입할 수 있는지를 파악하고자 큐싱 탐지 웹 서비스를 구축하였다.

본 시스템은 모던 웹 프레임워크인 React를 사용하여 사용자 인터페이스를 구축하고, Python의 Flask 프레임워크를 기반으로 한 RESTful API 서버를 통해 Back-End 로직을 처리한다. 사용자는 웹 애플리케이션을 통해 QR 코드를 스캔하고, 스캔 된 데이터가 URL인 경우 이를

Back-End 서버로 전송하여 실시간으로 안전성 검사를 수행한다. Flask 서버는 전송된 URL을 분석하고, 해당 URL의 특성을 추출한 후 머신러닝 모델의 입력으로 사용한다. 기존 학습 데이터셋을 구축할 때 사용된 URL 추출 특징들을 입력받은 URL에서도 추출하여 표 1의 형식으로 모델에 입력한다.

모델로부터 예측 작업 후, 반환된 결과 값은 0(악성) 또는 1(정상)로, 이를 바탕으로 Front-End 단위에서 처리하게 된다.

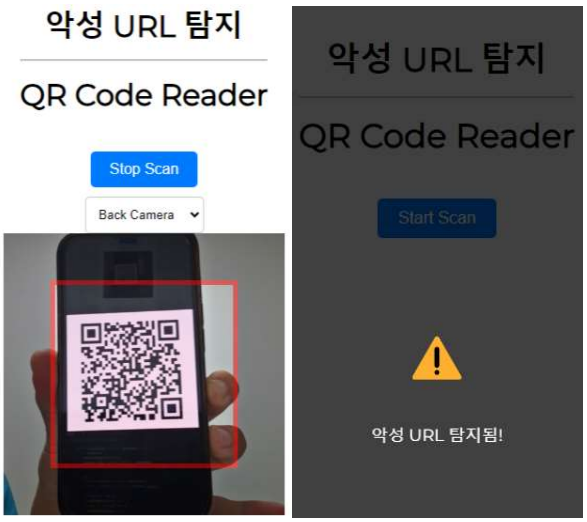


그림. 악성 URL 탐지 QR코드 스캐너 실행 결과
(좌) 카메라를 통한 QR코드 인식 화면 (5)
(우) 악성 QR코드 탐지 후 경고 화면 (6)

악성 URL이 탐지된 경우, 시스템은 사용자 인터페이스에 경고 오버레이를 표시한다. 이 오버레이는 사용자가 악성 URL에 접근하는 것을 방지하기 위한 경고를 목적으로 한다. 반면, URL이 안전한 것으로 판별될 경우 자동으로 해당 URL로 Redirection 되어 원하는 콘텐츠에 접근할 수 있게 된다.

본 논문에서 개발한 악성 URL 탐지 모델을 기반으로 하여 실제 성능과 악성 QR코드를 탐지하는 목적에 부합하는지를 확인하고자 모델의 학습 데이터셋에 포함되지 않은 새로운 악성 URL로 제작한 테스트용 QR코드 100개를 기반으로 하여 실제 상황에 비슷한 환경으로 실험을 진행하였다. 실험 결과, 약 97%의 정확도를 기반으로 악성 URL 탐지에 성공하였다.

## Ⅶ. 결론

본 논문에서는 악성 URL 및 큐싱 공격을 탐지하기 위한 목적으로, 머신 러닝을 활용한 자동화된 모델 학습 접근법을 중심으로 연구를 진행하였다. 특히, AutoML 기술을 적용하여 다양한 머신 러닝 모델들 중에서 최적의 모델을 선정하고, 해당 모델을 학습시키는 과정에 초점을 맞추었다. 이 과정에서 RandomForest, ExtraTrees, WeightedEnsemble 모델이 높은 성능을 보여주었다.

모델의 학습 과정에서는 악성 URL과 정상 URL을 구분하기 위해 URL의 구조적 특성과 복잡성을 포함한 다양한 특징들을 추출하고 이를 기반으로 학습 데이터셋을 구성하였다. 이러한 고도화된 특징 추출 방법은 모델이 URL의 세밀한 차이를 학습하여 악성과 정상을 보다 정확하게 분류할 수 있도록 도왔다.

학습된 모델의 성능 평가에서는 정확도와 F1-Score를 중심으로 성능을 측정하였고, 특히 RandomForestEntr 모델이 뛰어난 성능을 보임으로써 본 연구의 학습 방법론과 모델 선택의 타당성을 입증하였다. 이러한 평가 결과는 AutoML을 통해 최적화된 모델이 악성 URL 탐지 문제에 있어 높은 정확도를 도출한다는 것을 의미한다.

이어진 연구 단계에서는 학습된 악성 URL 탐지 모델을 기존 QR 코드 리더기와 결합하여, 사용자가 QR 코드를 스캔할 때 실시간으로 URL의 안전성을 평가하고 위험한 URL을 효과적으

로 식별할 수 있는 시스템을 구현하였다. 구현된 시스템의 실제 환경에서의 테스트 결과, 높은 정확도를 바탕으로 악성 URL을 성공적으로 탐지하였으며, 이는 본 논문에서 제안한 기술이 실제 상황에서도 효과적으로 작동할 수 있음을 보여준다. 따라서, 본 연구는 악성 URL 탐지 기술뿐만 아니라, 큐싱 공격의 대응 방안 중 하나임을 증명하였다.

## REFERENCES


[1] 'QR코드 활용' 신종 금융사기 '큐싱' 주의보(2015), https://cm.asiae.co.kr/article/2015052709110452713, (accessed Jan., 08, 2024).

[2] What is a phishing attack? by CloudFlare(2024), https://www.cloudflare.com/ko-kr/learning/access-management/phishing-atack, (accessed Jan., 08, 2024).

[3] Woman receives $788 Walmart charge after scanning QR code in downtown Lafayette parking lot(2023), https://www.klfy.com/local/lafayette-parish/woman-receives-788-walmart-charge-after-scanning-qr-code-in-downtown-lafayette-parking-lot/,(accessed Jan., 09, 2024).

[4] 김은결, 김보람, 권소연, 김유빈, and 이광재, "A Development of a Deep Learning-based Qshing Detection App using DGA and WHOIS Information," *제어로봇시스템학회 국내학술대회 논문집,* 577-578쪽, 2023년 06월

[5] 한국인터넷진흥원_피싱사이트 URL(2022),https://www.data.go.kr/data/15109780/fileData.do, (accessed Jan., 12, 2024).

[6] Malicious URLs dataset(2021), https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset/data, (accessed Jan., 12, 2024).

[7] Chae-rim Han, Su-hyun Yun, Myeong-jin Han, and Il-Gu Lee, "Machine Learning -Based Malicious URL Detection Technique," Journal of the Korea Institute of Informati
on Security & Cryptology, vol. 32, no. 3, pp. 555-564, 2022.

[8] 한채림, 윤수현, 한명진, and 이일구, "머신러닝 기반 악성 URL 탐지 기법," *정보보호학회 논문지,* 제32권, 제3호, 555-564쪽, 2022년 6월

[9] Zöller, M.A., Huber, M.F., "Benchmark and survey of automated machine learning frameworks," *Journal of Artificial Intelligence Research 70,* pp. 409-472. 2021.

[10] 장성민, 김준학, 권희정, 오은희, 서창진, "RandomForest와 XGBoost를 이용한 악성코드 탐지 시스템 개발," *대한전기학회 학술대회 논문집,* 제주, 대한민국, 2023년 5월


### 저 자 소 개


**김동영(학생회원)**

2023년 ~ 현재 경북소프트웨어고등학교 소프트웨어개발과 재학중
2024년 한국정보기술연구원 차세대 보안리더 양성 프로그램 화이트햇스쿨 WHS 1기 수료생

<주관심분야 : 사이버보안, 보안제품개발, DevSecOps, 인공지능>

**황기성(학생회원)**

2023년 ~ 현재 경북소프트웨어고등학교 인공지능소프트웨어과 재학중

<주관심분야 : AI, Computer Vision, Multi Model>